第二周

1. 关键字，变量，基础语法
2. Java语句（复合，条件，多分支和跳转）
3. Java集合框架
4. 抽象，继承，多态的编写（小型demo）有基础的可以结合各种设计模式实现

常用关键字：

JAVA的关键字都是小写的

1、JAVA的保留关键字（2个）

     const——常量，常数：用于修改字段或局部变量的声明。

     goto——转到：指定跳转到标签，找到标签后，程序将处理从下一行开始的命令

2、访问修饰符的关键字（3个）

     public （公有的）：可跨包

     protected (受保护的)：当前包内可用

     private (私有的)：当前类可用

3、定义类、接口、抽象类和实现接口、继承类的关键字、实例化对象（6个）

     class (类)：public class A(){}花括号里是已实现的方法体，类名需要与文件名相同

     interface (接口)：public interface B(){}花括号里有方法体，但没有实现，方法体句子后面是英文分号;结尾

     abstract (声明抽象)：public abstract class C(){}介于类与接口中间，可以有，也可以没有已经实现的方法体

     implemenst (实现)：用于类或接口，实现接口public class A interface B(){}

     extends (继承)：用于类继承类public class A extends D(){}

     new (创建新对象)：A a=new A();A表示一个类

4、包的关键字（2个）

     import (引入包的关键字)：当使用某个包的一些类时，仅需要类名，然后使用ctrl+shift+o或者选定类名（类或属性或方法）按住ctrl+单击，即可自动插入类所在的包

     package (定义包的关键字)：将所有相关的类放在一个包类以便查找修改等

5、数据类型的关键字（12个）

     byte (字节型)：8bit

     char (字节型)：16bit

     boolean (布尔型)：--

     short (短整型)：16bit

     int (整型)：32bit

     float (浮点型)：32bit

     long (长整型)：64bit

     double (双精度)：64bit

     void (无返回)：public void A(){}其他需要反回的经常与return连用

     null (空值)

     true (真)

     false (假）

6、条件循环（流程控制）（12个）

     if (如果) ：if（条件语句｛执行代码｝如果条件语句成立，就开始执行｛｝里面的内容

     else (否则，或者) ：常与if连用，用法相同：if(...){...}else{...}

     while (当什么时候)：while（条件语句）｛执行代码｝

     for（满足三个条件时）：for(初始化循环变量；判断条件；循环变量值｛｝

    switch (选择结构)：switch(表达式）｛case 常量表达式1：语句1；...case 常量表达式2；语句2；default:语句；｝default就是如果没有匹配的case就执行它，default并不是必须的。case后的语句可以不用大括号。

    case (匹配switch的表达式里的结果) ：同上

    default (默认)： default就是如果没有匹配的case就执行它， default并不是必须的

    do (运行) ：通长与while连用

    break (跳出循环)：直接跳出循环，执行循环体后的代码

    continue (继续) ： 中断本次循环，并开始下一轮循环

    return (返回) ：return 一个返回值类型

    instanceof(实例)：一个二元操作符，和==、>、<是同一类的。测试它左边的对象是否是它右边的类的实例，返回boolean类型的数据

7、修饰方法、类、属性和变量（9个）

    static(静态的)：属性和方法都可以用static修饰，直接使用类名、属性和方法名。只有内部类可以使用static关键字修饰，调用直接使用类名、内部类类名进行调用。static可以独立存在

    final(最终的不可被改变)：方法和类都可用final来修饰；final修饰的类是不能被继承的；final修饰的方法是不能被子类重写。常量的定义：final修饰的属性就是常量

    super(调用父类的方法)：常见public void paint(Graphics g){super.paint(g);...}

    this(当前类的父类的对象)：调用当前类中的方法（表示调用这个方法的对象）this.addActionListener(al):等等

    native(本地)

    strictfp(严格，精准)

    synchronized(线程，同步)：一个时间内只能有一个线程得到执行。另一个线程必须等待当前线程执行完这个代码块以后才能执行该代码块

    transient(短暂)

    volatile(易失)

 8、错误处理（5个）

    catch(处理异常)：（1）try+catch 程序流程是：运行到try块中，如果有异常抛出，则转到catch块去处理。然后执行catch块后面的语句

                                 （2）try+catch+finally 程序流程是：运行到try块中，如果有异常抛出，则转到catch垮，catch块执行完毕后，执行finally块的代码，再执行finally块后面的代码。如果没有异常抛出，执行完try块，也要去执行finally块的代码。然后执行finally块后                                                 面的语句

                                 （3）try+finally  程序流程是：运行到try块中，如果有异常抛出，则转到finally块的代码。

   try(捕获异常)

  finally（有没有异常都执行）

  throw(抛出一个异常对象)：一些可以导致程序出问题，比如书写错误，逻辑错误或者是api的应用错误等等。为力防止程序的崩溃就要预先检测这些因素，所以java使用了异常这个机制

                                            在java中异常是靠“抛出”  也就是英语的“throw”来使用的，意思是如果发现到什么异常的时候就把错误信息“抛出”

  throws(声明一个异常可能被抛出)：把异常交给他的上级管理，自己不进行异常处理

9、其他（2个）

   enum(枚举，列举，型别)

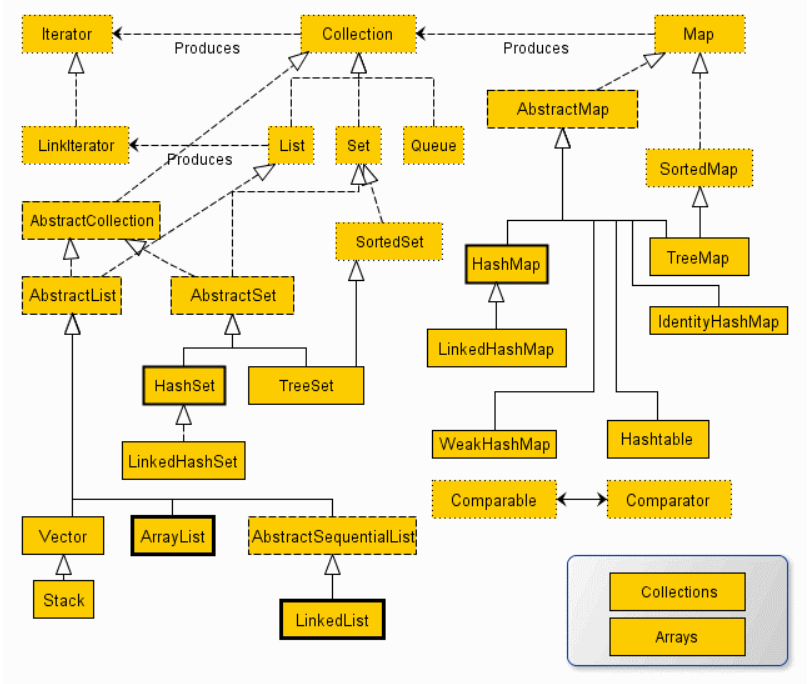
   assert(断言)

变量：

Java语言支持的变量类型有：

* 类变量：独立于方法之外的变量，用 static 修饰。
* 实例变量：独立于方法之外的变量，不过没有 static 修饰。
* 局部变量：类的方法中的变量。

Java集合框架



从上面的集合框架图可以看到，Java 集合框架主要包括两种类型的容器，一种是集合（Collection），存储一个元素集合，另一种是图（Map），存储键/值对映射。Collection 接口又有 3 种子类型，List、Set 和 Queue，再下面是一些抽象类，最后是具体实现类，常用的有 [ArrayList](https://www.runoob.com/java/java-arraylist.html)、[LinkedList](https://www.runoob.com/java/java-linkedlist.html)、[HashSet](https://www.runoob.com/java/java-hashset.html)、LinkedHashSet、[HashMap](https://www.runoob.com/java/java-hashmap.html)、LinkedHashMap 等等。

集合框架是一个用来代表和操纵集合的统一架构。所有的集合框架都包含如下内容：

* **接口：**是代表集合的抽象数据类型。例如 Collection、List、Set、Map 等。之所以定义多个接口，是为了以不同的方式操作集合对象
* **实现（类）：**是集合接口的具体实现。从本质上讲，它们是可重复使用的数据结构，例如：ArrayList、LinkedList、HashSet、HashMap。
* **算法：**是实现集合接口的对象里的方法执行的一些有用的计算，例如：搜索和排序。这些算法被称为多态，那是因为相同的方法可以在相似的接口上有着不同的实现。